



#ШПАРГАЛОЧКИ

# СОЗДАНИЕ САЙТОВ FRONT-END РАЗРАБОТКА

## Дополнительный уровень

Материалы подготовлены отделом методической разработки

Больше полезных материалов и общения в нашем комьюнити  
в Telegram: [https://t.me/hw\\_school](https://t.me/hw_school)





# Внедрение функции subscribe, погружение в ООП



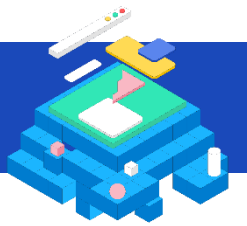
# subscribe

Есть способ обойти цикличную зависимость без использования дополнительного скрипта. Код из `render.js` нужно вернуть в `index.js`, в этом файле функция перерисовки будет вызываться один раз (для первого отображения страницы):

**`rerenderTree(state)`**

А в **`state.js`** нужна новая функция **`subscribe`**. Она принимает параметр (наблюдателя) **`observer`** и записывает его в **`rerenderTree`**.

```
export let subscribe = (observer) => {  
  
    rerenderTree = observer  
  
}
```



# subscribe

Остается импортировать и вызвать эту функцию в **index.js**:

**subscribe(renderTree)**

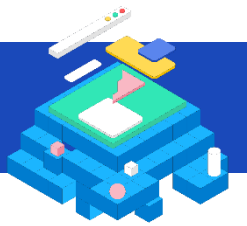
Теперь перерисовкой приложения занимается **subscribe**.



# Объекты и методы

**Метод** - это функция, созданная внутри объекта и умеющая работать с данными этого объекта:

```
let person = {  
  name: "Vasya",  
  age: "15",  
  sayName() {  
    console.log(this.name);  
  }  
}
```



# Объекты, методы и this

Важно, что обращаться к полям (например, `name`) в методе нужно через специальное слово **this** и точку - **this.name**.

**this** обозначает, что метод работает с полем **name**, принадлежащем именно этому объекту.

А вот извне объекта (из основной программы) доступ к его полям и методам происходит через имя объекта:

```
console.log(person.age);
```

```
person.sayName();
```



# Приватность, set и get

Не всю информацию из объекта стоит делать доступной извне. Поля, доступ к которым должен быть ограничен, можно сделать защищенными, добавив к их имени нижнее подчеркивание:

```
let person {  
    _name = "Vasya",  
}
```

А для доступа к таким данным в объект добавляются специальные методы - **геттеры** (возвращающие значение поля) и **сеттеры** (устанавливающие новое значение).

Важно: JavaScript не ограничивает доступ к защищенным полям. Поэтому добавление нижнего подчеркивания - это просто способ сообщить другим разработчикам, что это поле должно быть приватным.



# Приватность, set и get

```
let person {  
  
    _name = "Vasya",  
  
    getName() {  
  
        return this._name;  
  
    }  
  
    setName(name) {  
  
        this._name = name;  
  
    }  
  
}
```